Matrix: Jurnal Manajemen Teknologi dan Informatika
Volume 11 Issue 1 Year 2021 Pages 1-10

DOI: http://dx.doi.org/10.31940/matrix.v11i1.2335
URL: http://ojs.pnb.ac.id/index.php/matrix

livE (onLine – java Exercise) Java Programming Language Learning System for lab and online test

Mamluatul Hani'ah [1]*, Yogi Kurniawan [2], Imam Fahrur Rozi [3]

[1] Politeknik Negeri Malang, Indonesia
[2] Universitas Brawijaya, Indonesia
[3] Politeknik Negeri Malang, Indonesia

*Corresponding Author: *mamluatulhaniah@polinema.ac.id*

Abstract: Java programming language is one of the most popular programming languages among IT developers. There are common problems in the beginning learning process such as the installation and configuration process of java software like JDK (Java Development Kit). During the online class session, it takes a lot of time and a large internet data packet to solve this problem. This research proposes a website-based system called livE (online - java Exercise). Students can practice coding and also do exams in real-time, simply by using a web browser and the internet. If lecturers want to make corrections, lecturers do not need to download student answer files. They can directly run the program code from student answers in the livE system. This indicates that the system implementation provides process efficiency in the student's job sheet exercises and lecturer's corrections. The result from the test that has been carried out shows that livE is running as expected. Usability testing shows that users are "totally agree" with system quality.

Keywords: Java Programming Language, livE, lab, online examination system

History Article: Submitted 28 January 2021 | Revised 10 February 2021 | Accepted 4 March 2021

How to Cite: M. Hani'ah, Y. Kurniawan, and I. F. Rozi, "livE (online – java Exercise) Java Programming Language Learning System for Lab and Online Test," Matrix: Jurnal Manajemen Teknologi dan Informatika, vol. 11, no. 1, pp. 1–10, 2021.

Introduction

Information technology is developing so rapidly. It cannot be denied that in Indonesia, information technology had a lot of influence on many sectors. Of course, information technology is commonly used to improve education [1][2], economy [3][4], transportation [5], health [6][7], agriculture [8] and data security [9]. Developing information technology cannot be separated from the role of IT developers. One of the programming languages that IT developers learn is the Java programming language.

Java programming language is one of the most popular programming languages among IT developers. This programming language is usually used as basics programming education material on informatic studies programs at the college. Of course, the Java programming language is fundamental for students because it will be the opening gate for informatics engineering students to learn other programming languages. One way to improve student skills and abilities is by doing practical work or exercise using the Java programming language.

In offline class, many students experience problems. The problem occurs when students perform installation and configuration process on practical devices like installation and configuration of Java JDK (Java Development Kit). It is often happening in the beginning semester, especially students who come from non vocational high school and are not familiar with computer programming. Students that fail to install and configure the practical supporting software, cannot perform an experiment and practical exercises. To resolve these problems, the lecturer must provide assistance and direction to the students.

However, during the online lecture session, lecturers could not directly assist students. Meanwhile, if the problem is solved by lecturers using video conferencing, it will take a relatively long video conference time and spend relatively many internet data packages. When a student does not immediately repair the problem, the student will not be able to do the experiment steps or exercises from the practical's job sheet.

Several studies have been conducted by other researchers, developing systems for learning and coding in real-time programming languages. Research by M. Guo et al. developed an automatic assessment for learning the introduction of the Java programming language for computer science students [10]. The system built aims to make students learn interactively and in real-time. Furthermore, research conducted by A. N. Kumar made an online compiler as a system designed for learning the C ++ programming language [11]. The system also provides animations to assist students in learning se-mantic pointers in the C ++ programming language.

Research by N. Funabiki et al. designed a website-based application, called the Java Programming Learning Assistant System (JPLAS) [12]. It proposes to improve student skills in the Java programming language. JPLAS was designed for the teacher so the teacher can provide questions and program code. Then students can answer questions from the lecturer by creating program code that can be compiled directly on the JPLAS web.

This research proposes a website-based system called "livE (online - java Exercise)" to improve the learning process using Java programming language. Students can work on job sheets and question exercises through the livE system (online - java Exercise). Lecturers can use the livE system to conduct online exams that require students to practice coding java programming language directly. Students can answer the exam by coding in the livE system. Furthermore, students can run the program code to find out the output of the program code. This system also provides a reporting feature, how many times a student-run program code to answer a case study/question. The benefit of this system is students can practice coding, and also they can carry out practical tests in real-time, just using a web browser and the internet. Lecturers will also find it easy by using this system. If lecturers want to make corrections, lecturers do not need to download student answer files. They can directly run the program code from student answers in the livE system. This indicates that the system implementation provides process efficiency in the student's job sheet exercises and lecturer's corrections.

Methodology

Figure 1 shows the process/stages in this research include literature review and observation, system requirements analysis, systems design, build software based on the design, and last is software testing. Literature review and observation is to study the latest problems and research. System requirements analysis is to find out the needs of the system.

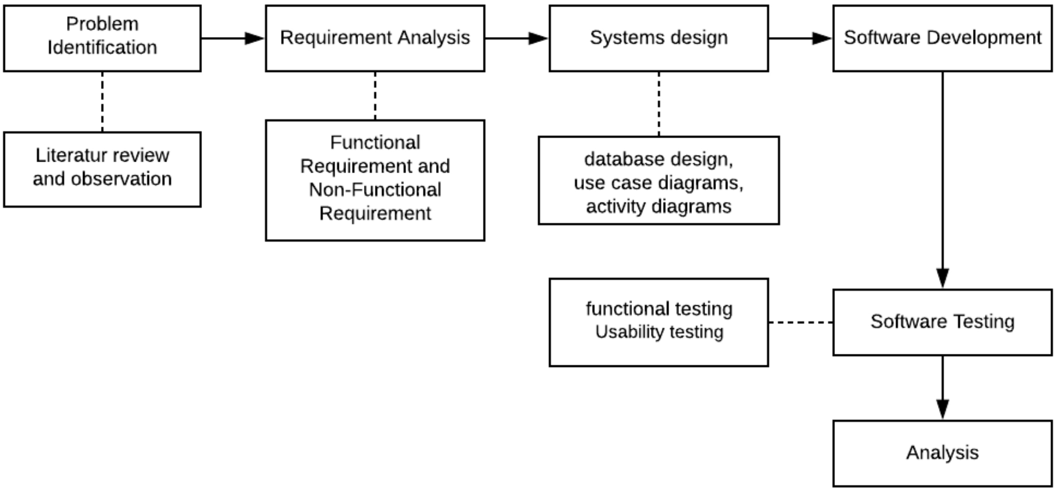

Problem Identification
Requirement Analysis
Systems design
Software Development
Literatur review and observation
Functional Requirement and Non-Functional Requirement
database design, use case diagrams, activity diagrams
functional testing Usability testing
Software Testing
Analysis

Figure 1. Stages of system development

The first stage is to identify the problem through literature review and observation. Literature reviews are concerned with an understanding of the theoretical and technical basics of this study. The literature review process resulted in the theory used as well as previous studies. Furthermore, Researchers find research problems through observation by surveyed 31 students who have used the Java programming language in the first semester. From this survey, it turns out that 41.9% (13) of students experience difficulties in installing and configuring Java software such as JDK. Apart from solving software installation and configuration problems, the "livE" system hopefully makes lecturers easier. Lecturers can check student answers in practice exam using Java programming language.

The requirements analysis stage is the second stage in this study. At this stage, the functional requirements of the system are determined that consist of hardware and software required by the system also be analysed that functional requirements and hardware and software requirements.

Functional requirements consist of:

1. Users of the system, divided into three: admin, lecturers, and students.
2. All users can login to the system.
3. All users can make changes to data such as username, name, and password, especially student users can make changes to data from the school.
4. Lecturers can add exams/job sheet list, add exams/job sheet questions, add exams/job sheet participants, and give score to participants.
5. Students can take exams/ job sheet practice and can see the result of the exam/ job sheet results.

Hardware and software requirements consist of:

1. Hardware requirements. The computer specifications used in the development of this system are computers with an Intel (R) Core (TM) i5-3230M CPU @ 2.60GHz 8.00GB RAM.
2. Software requirements. The software environments used in the development of this system are Centos 8 Operating System, Apache Web server, Database Management System (DBMS) MariaDB, PHP Programming Language.

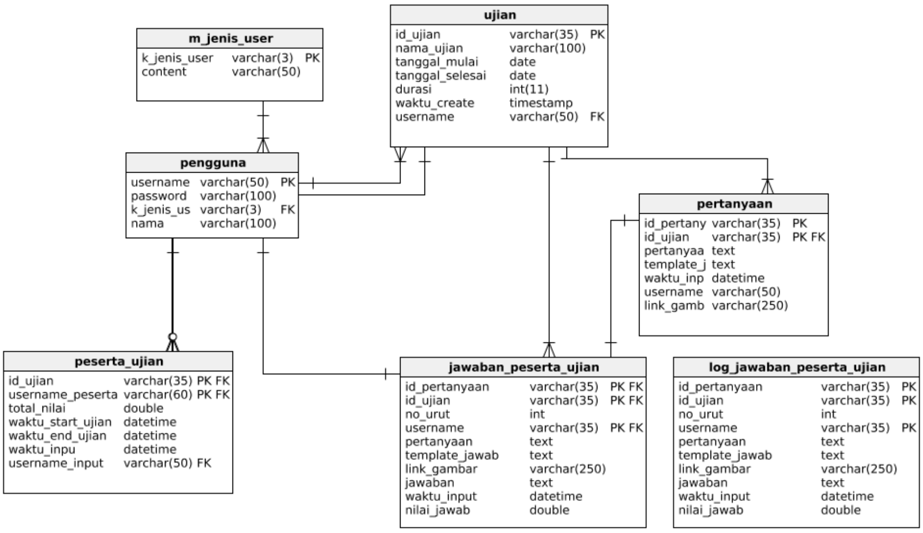


Figure 2. Database design of livE system

The third stage is the system design stage. This stage is needed to find a picture of what features are needed to support the function of the system. In this design process, it is done by creating the required database design, making use case diagrams to determine the role of each user, and activity diagrams to determine the flow of each feature. The database required in this study is illustrated with the physical diagram in Figure 2. In this study, seven tables were used, namely the *m_jenis_user*, *ujian*, *pengguna*, *pertanyaan*, *jawaban_peserta_ujian*, *peserta_ujian*, *log_jawaban_peserta_ujian*. The next design is making use case diagrams which can be seen in Figure 3. Use case diagrams are closely related to the scenario. Scenarios must be able to explain the complete and valid functionality of the system [13]. The interaction between users (actors) in the livE system is clearly illustrated in the use case diagram. The last design process is making an activity diagram which is used to describe the workflow of a system. Following the use case that has been made, there are 8 activity diagrams for the livE system.

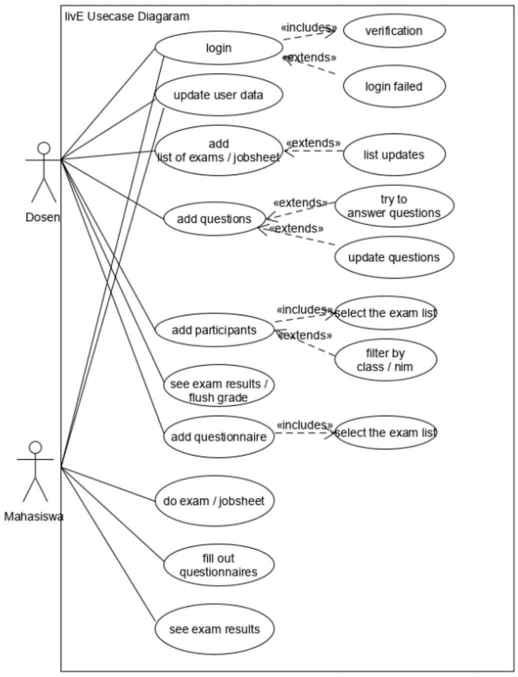


Figure 3. Use case diagrams of livE system

The fourth stage is making the system. In this process, the system coding carried following the design. livE implemented using JDK 14, Apache web server, Centos 8 operating system, DBMS (Database Management System) Maria DB 10.3, and the PHP programming language. When the user enters the Java program code in a web browser, These codes store on a text file. Then the text file will be compiled and executed using the Shell Function in PHP. After the execution is complete, the results store on the text file, and PHP will read them and return them to the user. The system architecture created is shown in Figure 4.

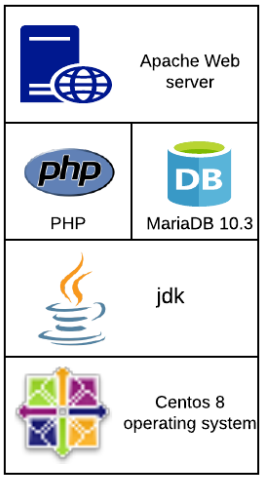


Figure 4. System architecture of livE

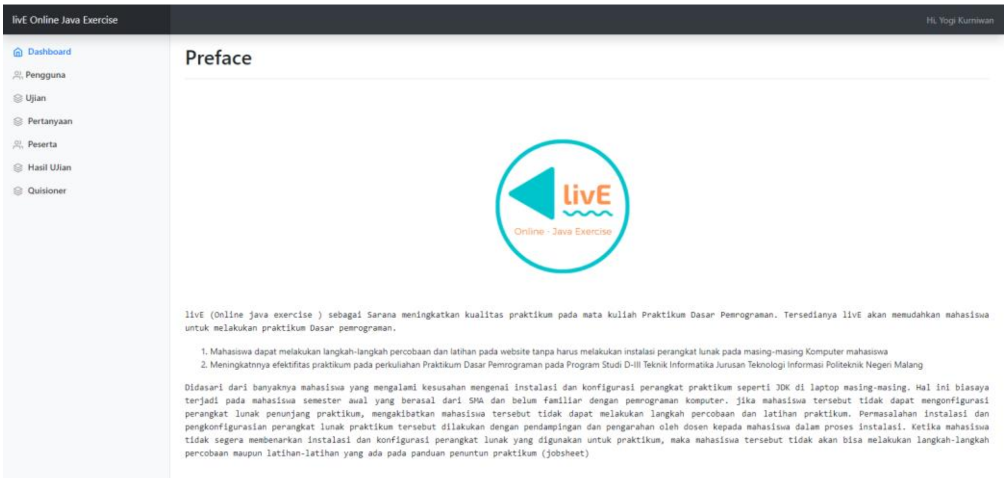
The fifth stage is system testing. The system is tested using functional testing and usability testing. Functional testing aims to test system features, whether these features are running well. This functional testing creates a test cases based on the requirement or design specification [14]. Meanwhile, usability testing measures the quality of a product or system-based experience of users when using the product [15]. This test aims to find out how good the system is, based on a questionnaire that has been filled in by the user [16].

Results and Discussions

Students can use the "livE" system when they want to do job sheets or take online exams when to practice coding in the Java programming language. This chapter will present the results of the system and also a discussion of system testing.

Results

The livE (online - java Exercise) can be used by students in lab experiments and carrying out online exams. So that students can practice coding in the Java programming language. Three types of users can interact with the system, the first user is a lecturer user, the second user is a student, and the last is a lecturer who is the administrator.

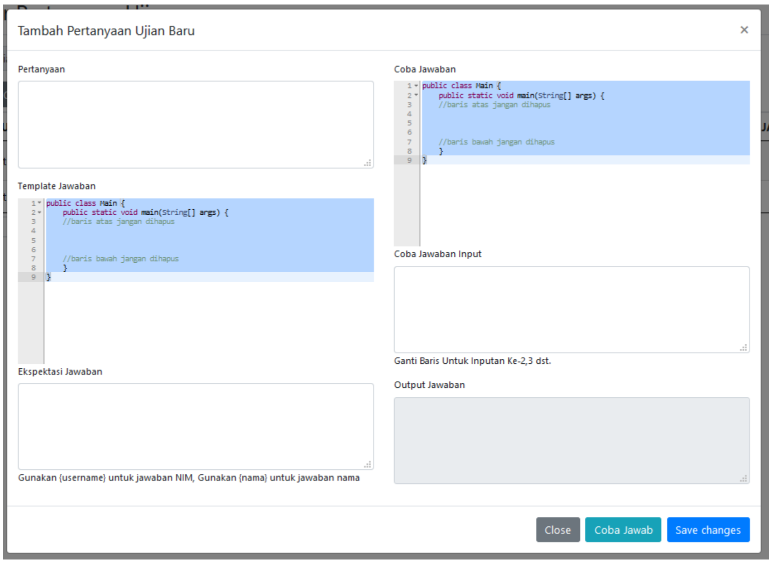


livE (Online java exercise) sebagai Sarana meningkatkan kualitas praktikum pada mata kuliah Praktikum Dasar Pemrograman. Tersedianya livE akan memudahkan mahasiswa untuk melakukan praktikum Dasar pemrograman.

1. Mahasiswa dapat melakukan langkah-langkah percobaan dan latihan pada website tanpa harus melakukan instalasi perangkat lunak pada masing-masing Komputer mahasiswa
2. Meningkatnnya efektifitas praktikum pada perkuliahan Praktikum Dasar Pemrograman pada Program Studi D-III Teknik Informatika Jurusan Teknologi Informasi Politeknik Negeri Malang

Didasari dari banyaknya mahasiswa yang mengalami kesusahan mengenai instalasi dan konfigurasi perangkat praktikum seperti JDK di laptop masing-masing. Hal ini biasaya terjadi pada mahasiswa semester awal yang berasal dari SMA dan belum familiar dengan pemrograman komputer. jika mahasiswa tersebut tidak dapat mengonfigurasi perangkat lunak penunjang praktikum, mengakibatkan mahasiswa tersebut tidak dapat melakukan langkah percobaan dan latihan praktikum. Permasalahan instalasi dan pengkonfigurasian perangkat lunak praktikum tersebut dilakukan dengan pendampingan dan pengarahan oleh dosen kepada mahasiswa dalam proses instalasi. Ketika mahasiswa tidak segera membenarkan instalasi dan konfigurasi perangkat lunak yang digunakan untuk praktikum, maka mahasiswa tersebut tidak akan bisa melakukan langkah-langkah percobaan maupun latihan-latihan yang ada pada panduan penuntun praktikum (jobsheet)

Figure 5. Lecturer dashboard



```
public class Main {
    public static void main(String[] args) {
    //baris atas jangan dihapus



    //baris bawah jangan dihapus
    }
}
```

Gunakan {username} untuk jawaban NIM, Gunakan {nama} untuk jawaban nama

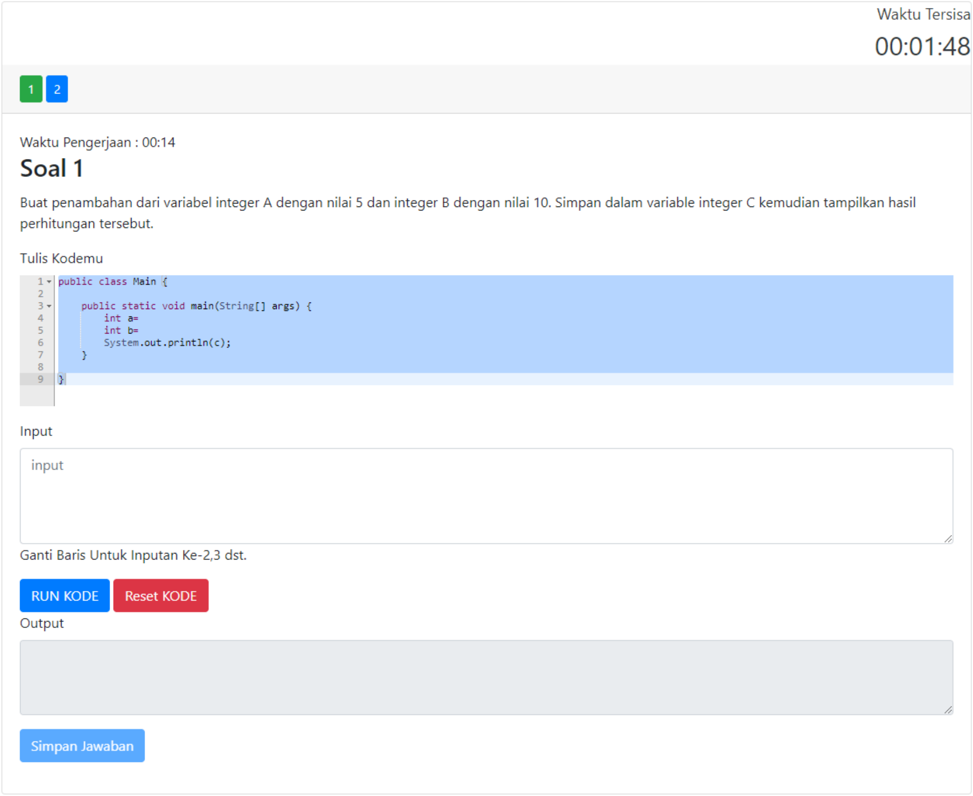
```
public class Main {
    public static void main(String[] args) {
    //baris atas jangan dihapus



    //baris bawah jangan dihapus
    }
}
```

Ganti Baris Untuk Inputan Ke-2,3 dst.

Figure 6. Add question interface

Figure 5 is the starting page when the lecturer successfully logs in. Lecturer users can make improvements to user data, schedule exams, add and subtract test participants, add, and subtract questions on the exam. They can also review the results of exams that have been done by students, provide scores on each exam participant's answer, and provide specific questionnaire questions on each exam.

Figure 6 is the interface when the lecturer adds new questions for both lab experiments and online exams. Experiment's command or question's description typed in the questions section. The answer template will always prefix with the Main class name. livE only execute the Main class. Every change on the answer template editor will be changing The Try Answer's editor. It makes easy for lecturers to try to answer the questions given. The Answer Expectations section contains the answers that the livE (Online - Java Exercise) considers correct. This section highly recommends being filled so that the assessment can be done automatically. Try Answer's Editors section can be used by the lecturer to verify the questions and answer templates that have been prepared. If there is input for the code, the try input answer section can be fulfilled.



Waktu Tersisa

00:01:48

Waktu Pengerjaan : 00:14

Soal 1

Buat penambahan dari variabel integer A dengan nilai 5 dan integer B dengan nilai 10. Simpan dalam variable integer C kemudian tampilkan hasil perhitungan tersebut.

Tulis Kodemu

```
public class Main {

    public static void main(String[] args) {
        int a=
        int b=
        System.out.println(c);
    }

}
```

Input

input

Ganti Baris Untuk Inputan Ke-2,3 dst.

Output

Figure 7. Exam/ practicum interface

For student users, the features provided are editing user data, viewing exams registered by lecturers to students, and seeing the exam scores. Figure 7 is the interface for the exam/practicum page. On the test page, there is time remaining in the top right corner. Below the remaining time, there are the question numbers from the exam. The active question number will be green and the other question numbers will be blue.

Discussions

This section discusses system testing performed on livE systems. System testing carried out in this study is functional testing and usability testing. Functional testing or BlackBox testing is one of the software testing methods. Functional testing is performed to ensure the product behaves according to functional requirements [14]. The results of functional testing in this study are shown in Table 1. Based on Table 1, it can be seen that the livE system has provided the expected results and has been running properly.

After the functional testing is complete, the next step is the usability test. A usability test in this study is done by giving a questionnaire to system users. livE users who fill out the questionnaire divided into two users, namely lecturers who teach Basic Programming courses and students who take fundamental programming courses. Lecturers need to fill in ten questions and were filled in by eight respondents, then the questionnaire to students was filled in by 40 respondents with six questions. Table 2 is the list of questions for the lecturer then Table 3 is the list of questions for students. User satisfaction ratings are measured using the Likert scale method [16]. In this study, the Likert scale used is five scales as shown in Table 4.

Table 1. Functional testing results

No.	Testing	Scenario	Status
1.	Login	The system has successfully received the data entered in the login menu. The data in the form is a username and password so that the user can access the system.	succeed
2.	Logout	The system has successfully accepted the logout process so system user can log out of the system.	succeed
3.	Update user data	The system displays user data changes according to user input.	succeed
4.	Add list of exams/ job sheet	The system displays the "Add a list of exams or job sheet" interface then the system will save input data and display that data. The system can also display a list of completed exams or job sheets.	succeed
5	Add questions	The system displays an interface for adding exam questions/job sheet questions then the system will save the user's input data and display it.	succeed
6	Add participants	In the selected exam/job sheets, the system can display a list of students who will add to the exam/job sheets. Then the system will save the list of participants.	succeed
7	Exam results	The system can display the predicted score by the system. Lecturers can make changes to that score. When the lecturer changes the score, the system will display the score according to the input.	succeed
8	Add questionnaire questions	The system can display the "add questionnaire" interface then the system will display the saved questions.	succeed
9	Start exams/ job sheet	The system displays the entire list of available exams/job sheets. The system will display exam questions when students press the start button.	succeed
10	Start coding	The system can display exam questions. The user can change the program code in the "write code" textbox.	succeed
11	Time-out	When the time for solving the questions runs out, the system automatically closes the questions. Then the system will redirect the user to the questionnaire page.	succeed
12	Fill in questionnaire	The system will display a list of questions and save the questionnaire answers.	succeed
13	Exam score	The system will display the score based on the test conducted by the user.	succeed

The questionnaire is processed using a Likert scale by calculating the percentage (%) index formula in Equation (1). Y is the highest Likert score multiplied by the number of respondents. Then the calculation of the interval using the formula in Equation (2) to classify response categories from users.

$$Percentage\ (\%) Index\ = \frac{SumTotal\ of\ the\ Product}{Y}\ x\ 100 \qquad (1)$$

$$Interval\ = \frac{100}{SumTotal\ of\ Skor\ (Likert)} \qquad (2)$$

Table 2. List of questions for lecturers

No	Questions
1	Is the "update user data" features easy to use to view data or make changes to user data?
2	Is the "Add list of exams/job sheets" feature on the livE System (online java exercise) easy to use?
3	Is the "Add questions" feature of the livE System (online java exercise) easy to use?
4	Is the "Add participants" feature on the livE system (online java exercise) easy to use?
5	Is the "Exam results" feature suit the lecturer's needs?
6	Is the "Add questionnaire " feature of the livE System (online java exercise) easy to use?
7	Overall, Is the livE system comfortable to use?
8	Overall, Is the livE System (online java exercise) easy to operate?
9	Overall, can this application be useful for the teaching team lecturers?
10	In your opinion, does the livE system (online java exercise) meet the student's requirement for Lab and Online Test?

Table 3. List of questions for students

No	Questions
1	Is the "exams/job sheets" feature on the livE system easy to use?
2	Can you easily code the java programming language using livE system?
3	Is the livE system help you in the Practice programming skills using java language?
4	Is the livE system meet your requirement?
5	Is the livE allow you to Practice programming skills using java language wherever you are?
6	Overall, is the livE easy to operate?

Table 4. Likert scale

Variable	Response Categories
1	Strongly disagree
2	Disagree
3	Neither agree nor disagree
4	Agree
5	Strongly agree

The interval value from the calculation using Equation (2) is 20 resulting in response categories shown in Table 5. Furthermore, Equation (1) produces a percentage usability testing. Lecturer users give 92.75%, student users give 86.58%. So, it can be concluded by Table 3 that the lecturers and students "totally agree" with the livE system.

Tabel 5. Classification based on interval value

Percentage	Response Categories
0% - 19.99%	Strongly disagree
20% - 39.99%	Disagree
40% - 59.99%	Neither agree nor disagree
60% - 79.99%	Agree
80% - 100%	Strongly agree

Conclusion

Students can use the "livE" to practice coding and also do exams in real-time. Lecturers can use the "livE" system too. Lectures can carry out online exams, which require students to practice coding using Java programming language directly. Based on the functional testing, the livE system can run according to the scenario. Meanwhile, usability testing shows that users "strongly agree" based on the user's experience interacting with the system. In the class sessions, the lecturer often wants to be able to identify student difficulties in coding. In computer programming, there are three kinds of errors: syntax errors, runtime errors, and logical errors. So, for further research, a livE system will be developed in which lecturers will get syntax errors/logic errors information.

References

[1] K. A. T. Indah and P. G. Sukarata, "Penjadwalan perkuliahan dengan metode metaheuristic ant colony optimization studi kasus Politeknik Negeri Bali," *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 9, no. 2, pp. 74–82, 2019.

[2] C. A. Cholik, "Pemanfaatan teknologi informasi dan komunikasi untuk meningkatkan pendidikan di Indonesia," *Syntax Literate: Jurnal Ilmiah Indonesia*, vol. 2, no. 6, pp. 21–30, 2017.

[3] I. G. P. F. P. Sudhana, I. K. Suwintana, A. D. Yulianthi, and I. G. A. O. Sudiadnyani, "Pengembangan aplikasi internet banking koperasi simpan pinjam berbasis web," *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 9, no. 3, pp. 89–95, 2019.

[4] B. Berlilana, R. Utami, and W. M. Baihaqi, "Pengaruh Teknologi Informasi Revolusi industri 4.0 terhadap perkembangan UMKM sektor industri pengolahan," *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 10, no. 3, pp. 87–93, 2020.

[5] D. A. Fitri, E. Rohadi, and E. Ekojono, "Optimasi biaya dan waktu jalur angkutan kota (angkot) Malang dengan metode ant colony berbasis perangkat android," *JIP (Jurnal Informatika Polinema),* vol. 3, no. 1, p. 20, 2016.

[6] M. Hani'ah, C. S. K. Aditya, A. Harto, and A. Z. Arifin, "Cortical bone segmentation using watershed and region merging based on statistical features," *Jurnal Ilmu Komputer dan Informasi*, vol. 8, no. 2, pp. 76–82, 2015.

[7] A. Ririd, M. Hani'ah, and I. Putri, "Analisis pertumbuhan balita menggunakan algoritma kmeans++ untuk mengetahui resiko obesitas," *SENTIA 2020*, vol. 12, no. 1, 2020.

[8] K. Karsid, A. W. Ramadhan, and R. Aziz, "Perbandingan kinerja mesin penetas telur otomatis dengan menggunakan kontrol on-off dan kontrol PWM," *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 8, no. 1, pp. 1–5, 2018.

[9] Y. Kurniawan and A. Tohari, "Pengembangan metode pengamanan berkas memafaatkan pewarnaan graf," *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 14, no. 2, pp. 190–197, 2016.

[10] M. Guo, T. Chai, and K. Qian, "Design of online runtime and testing environment for instant Java programming assessment," in *2010 Seventh International Conference on Information Technology: New Generations*, 2010, pp. 1102–1106.

[11] A. N. Kumar, "Data space animation for learning the semantics of C++ pointers," in *Proceedings of the 40th ACM technical symposium on computer science education*, 2009, pp. 499–503.

[12] N. Funabiki, Y. Matsushima, T. Nakanishi, K. Watanabe, and N. Amano, "A Java programming learning assistant system using test-driven development method," *IAENG International Journal of Computer Science*, vol. 40, no. 1, pp. 38–46, 2013.

[13] T. A. Kurniawan, "Pemodelan use case (UML): Evaluasi terhadap beberapa kesalahan dalam praktik," JTIIK: *Jurnal Teknolologi Informasi dan Ilmu Komputer*, vol. 5, no. 1, pp. 77–86, 2018.

[14] S. Nidhra and J. Dondeti, "Black box and white box testing techniques - A literature review," *International Journal of Embedded Systems and Applications*, vol. 2, no. 2, pp. 29–50, 2012.

[15] I. S. Junus, H. B. Santoso, R. Y. K. Isal, and A. Y. Utomo, "Usability evaluation of the student centered e-learning environment," *The International Review of Research in Open and Distributed Learning*, vol. 16, no. 4, pp. 62–82, 2015.

[16] B. Klug, "An overview of the system usability scale in library website and system usability testing," *Weave: Journal of Library User Experience*, vol. 1, no. 6, Apr. 2017.